LA-UR-21-21874
Approved for public release; distribution is unlimited.

Title: DRACO: An Overview

Author(s): Barber, John L.

Intended for: "Sandia Corrosion Monthly" Monthly Seminar at Sandia (Conducted in this case via Webex)

Issued: 2021-02-24

DRACO: An Overview

John L. Barber
Group T-1: Physics and Chemistry of Materials

I. Introduction

II. Capabilities

III. Example Applications

Los Alamos NATIONAL LABORATORY EST. 1943

Introduction

DRACO (**D**iffusion **R**e**AC**ti**O**n) is a diffusion and chemistry code designed to:

- Operate on 3D with an **unstructured grid** defining an arbitrary geometry of interacting parts.
- Generate its own **meshes** and use meshes created by **other software**.
- Model the transport of any number of diffusing quantities: Concentrations, pressures, temperature, etc.
- Allow **diffusion coefficients** to depend in an arbitrary way on concentration, temperature, position, time, etc.
- Model general **chemistry** between concentrations with arbitrary **reaction rates**.
- Allow arbitrary **initial conditions**, **boundary conditions**, and **sources/sinks**.
- Allow all of the above to be specified by the **user**.

Introduction

At its heart, DRACO is a **PDE solver** for sets of diffusion / reaction equations of the following form:

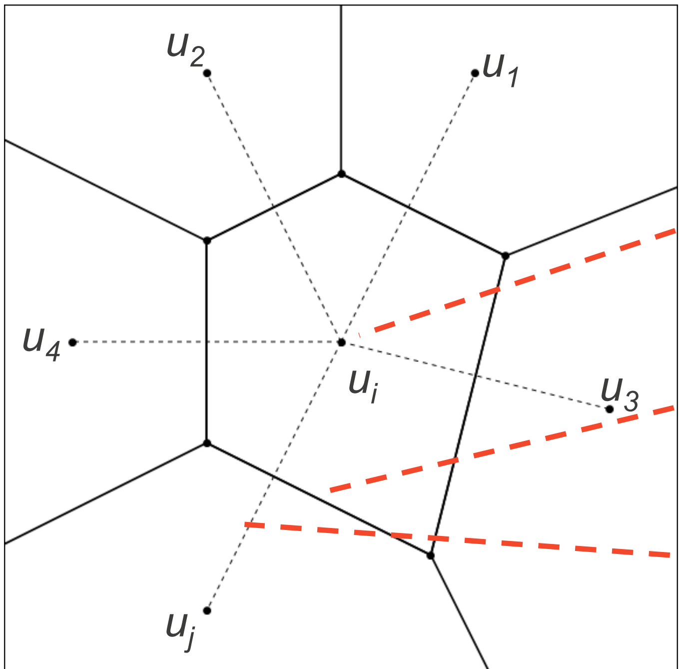
Scalar Diffusers
u, *v*, …

$$
\left\{
\begin{aligned}
\frac{\partial u}{\partial t} &= \overbrace{\vec{\nabla}\cdot\left(D_u \vec{\nabla} u\right)}^{\textbf{Diffusion}} + \left(\frac{du}{dt}\right)_{\mathrm{C}} + \overbrace{\left(\frac{du}{dt}\right)_{\mathrm{S}}}^{\textbf{Sources/Sinks}} \\
\frac{\partial v}{\partial t} &= \vec{\nabla}\cdot\left(D_v \vec{\nabla} v\right) + \underbrace{\left(\frac{dv}{dt}\right)_{\mathrm{C}}}_{\textbf{Chemistry}} + \left(\frac{dv}{dt}\right)_{\mathrm{S}} \quad \ldots
\end{aligned}
\right.
$$

Diffusion Coefficients
(May depend on concentration, temperature, position, time, etc.)

Introduction: Spatial Discretization

DRACO discretizes the **diffusion operator** using a tessellation of Voronoi cells around the points *i* of a mesh:



(2D Example of a Voronoi Tessellation)

$$\vec{\nabla} \cdot \left(D \vec{\nabla} u\right)_i \approx \frac{1}{V_i} \sum_j A_{ij} \, D_{ij} \, \frac{u_j - u_i}{d_{ij}}$$

$$D_{ij} = \frac{2}{1/D_i + 1/D_j}$$

V_i : Volume of cell *i*

A_{ij} : Area of the facet between cells *i* and *j*

d_{ij} : Distance between cells *i* and *j*

This is discretization scheme is 2nd order in the mesh size.

Introduction: Stability

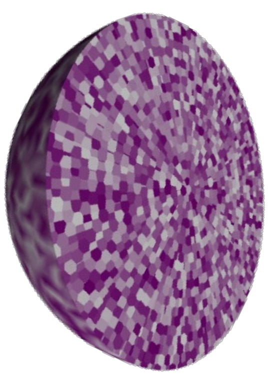
The stability of a DRACO simulation is determined by a **CFL number**:

$$CFL = \max_{ij} \frac{\Delta t\, A_{ij}\, D_{ij}}{2\, d_{ij}} \left(\frac{1}{V_i} + \frac{1}{V_j} \right) \propto \frac{D \Delta t}{\Delta x^2}$$

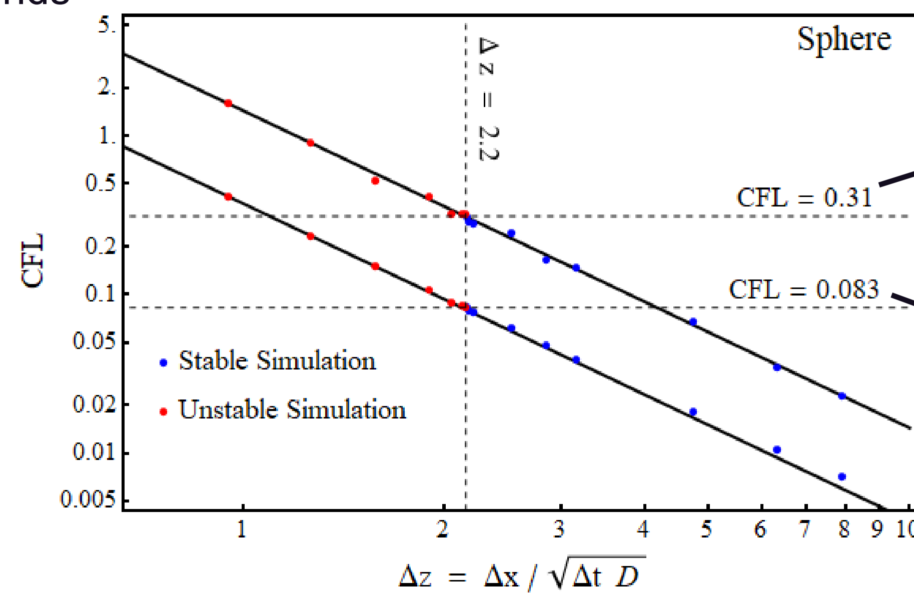
Time Step

Mesh Size

The exact criterion for stability depends on the mesh geometry.
Example: Diffusion in a sphere with Fibonacci meshing.



Simulations performed at different time steps and mesh sizes



Maximum CFL

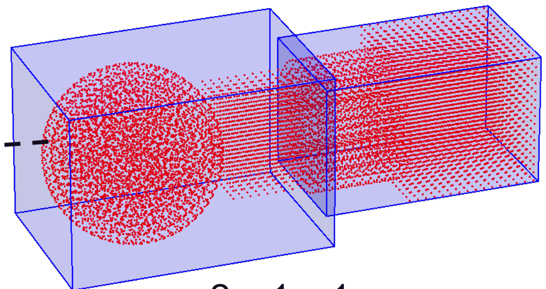
Average CFL

Introduction: Parallelization

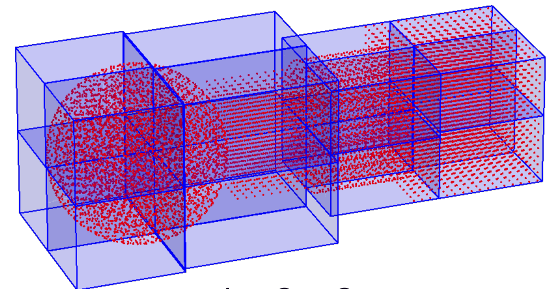
DRACO has been parallelized using the Message Passing Interface (MPI) library.
It uses a custom-designed algorithm to generate a box-based processor domain decomposition with reasonably-good load balancing:

Domain decomposition in a simple example system of 4 parts.

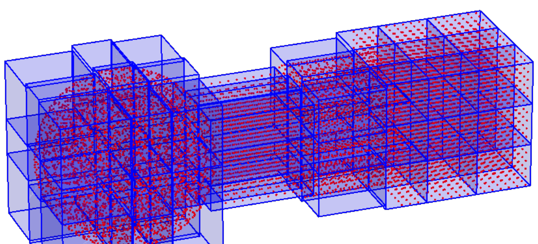
Red dots represent mesh points / grid points / elements.



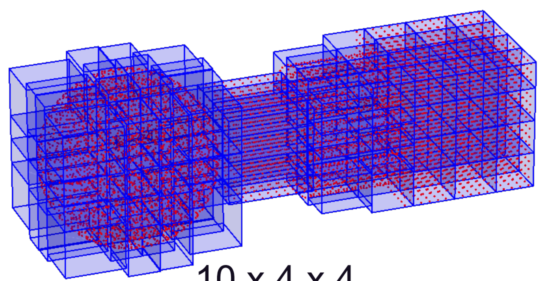
2 x 1 x 1



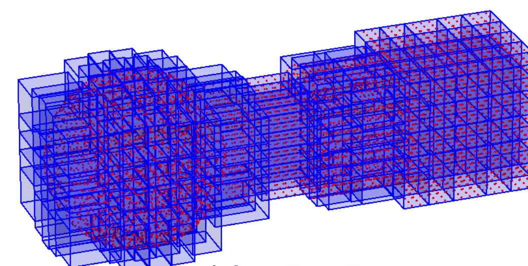
4 x 2 x 2



8 x 3 x 3



10 x 4 x 4



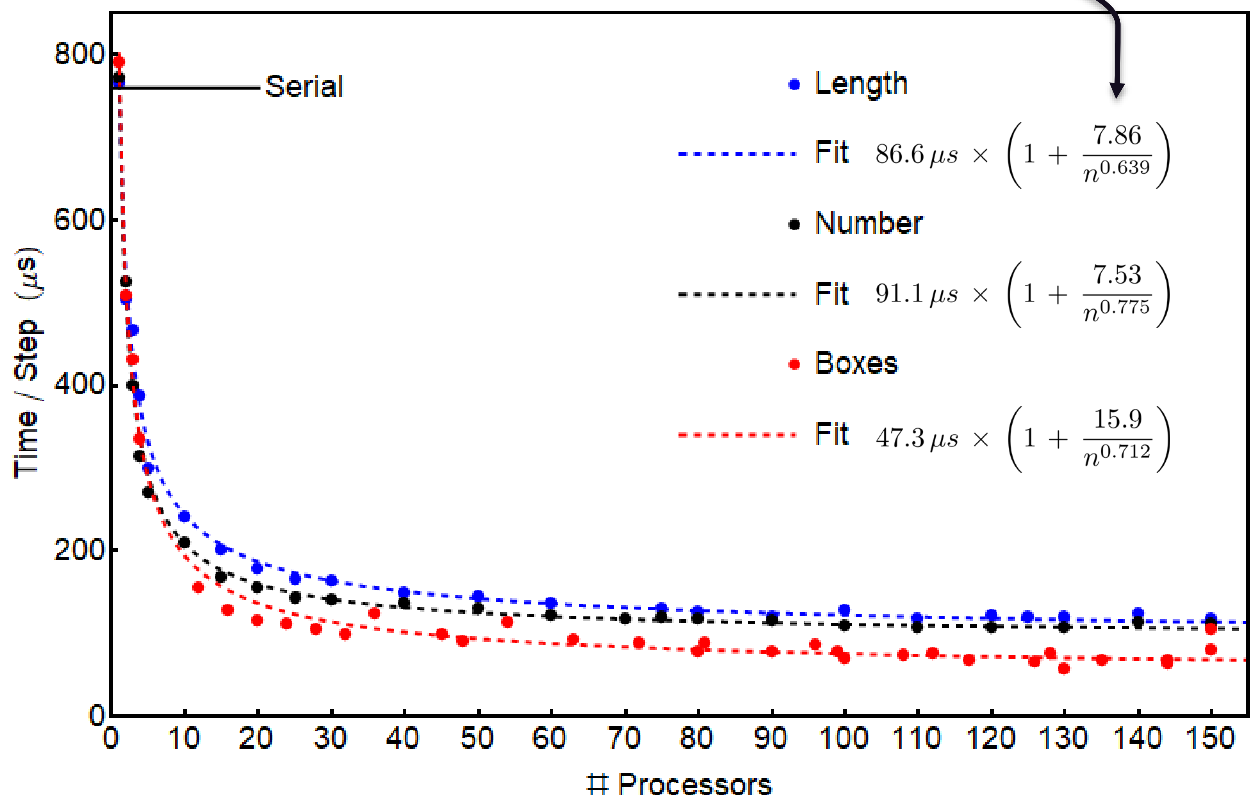
14 x 5 x 5

Introduction: Parallelization

Right: Scaling of the simulation time per step vs. the number of processors for an example simulation using several different domain decomposition schemes. (The one on the previous slide is labelled "Boxes" here.)

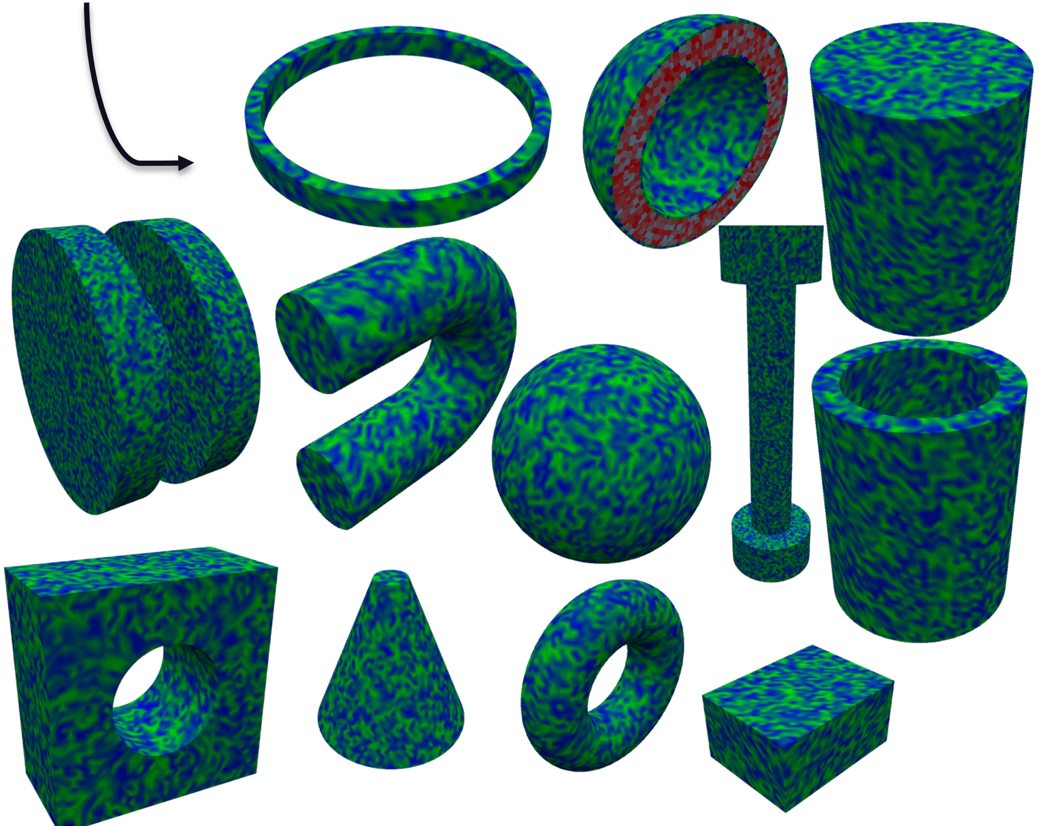
The eventual speed plateau happens due to communication overhead and the domain size approaching the mesh size.

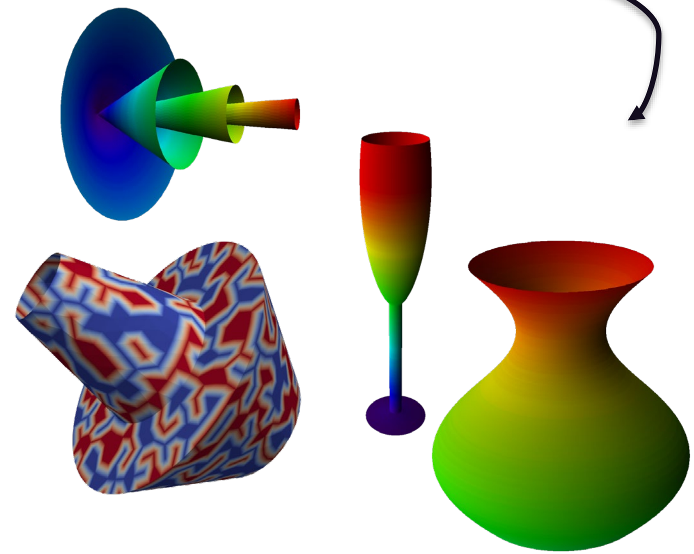
Power law scaling



Capabilities: Meshing and Rendering

DRACO can generate unstructured, Voronoi-based **meshes** for a collection of built-in **“native” shapes**, as well as for arbitrary, **radially-symmetric surfaces**.

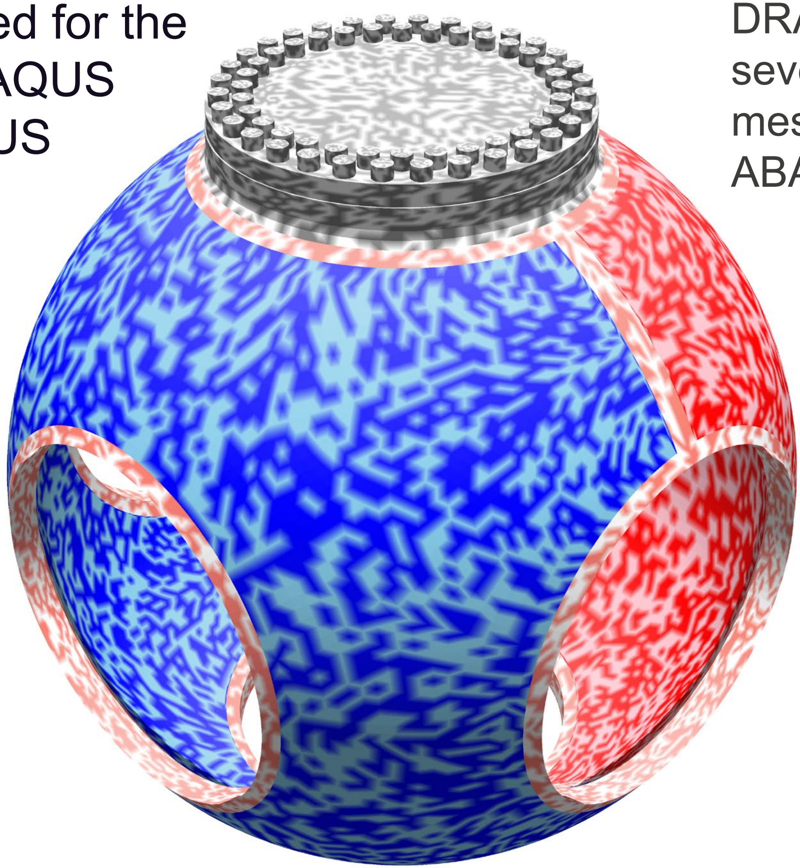




As illustrated here, DRACO can also perform **ray-tracing and rendering** calculations to generate images from simulations.

Capabilities: ABAQUS Meshes

DRACO can use meshes created for the popular finite element code ABAQUS by reading in an parsing ABAQUS input decks.

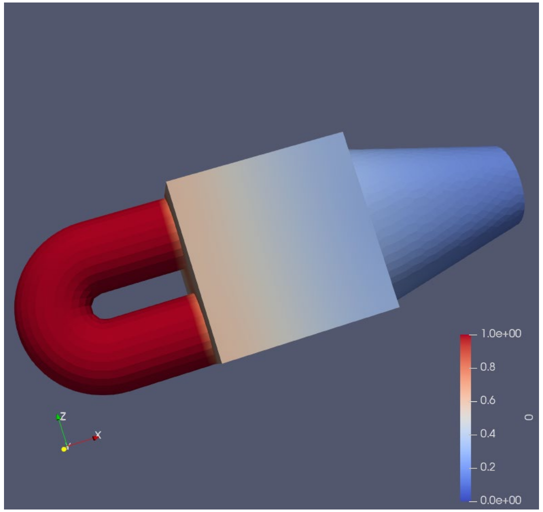


DRACO rendering of a several-dozen-part mesh loaded from an ABAQUS input deck.

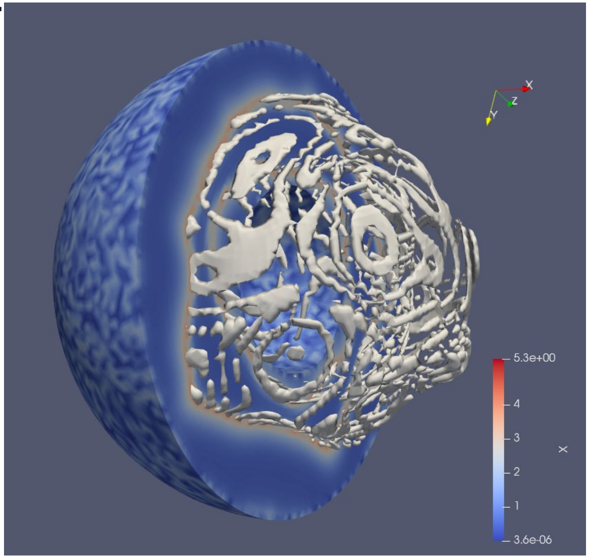
This allows the sharing of meshes between DRACO and other modeling efforts which use ABAQUS directly, e.g. for mechanical simulations.

Capabilities: VTK Files

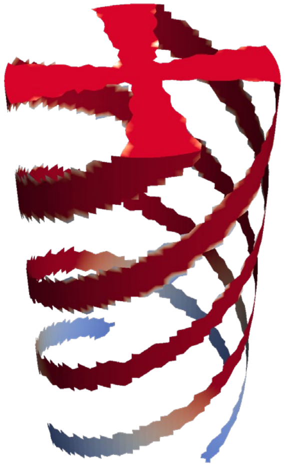
VTK (**V**isualization **T**ool**K**it) files are a popular file format for storing 3D geometries, which can be used and understood by many rendering software packages, e.g. ParaView. DRACO can periodically write VTK files describing the entire system during its calculations, allowing great flexibility for data visualization. Below, some examples of ParaView renderings using VTK files generated during DRACO simulations:



Diffusion in a test system of three parts



Oscillating chemistry and diffusion in a hollow sphere

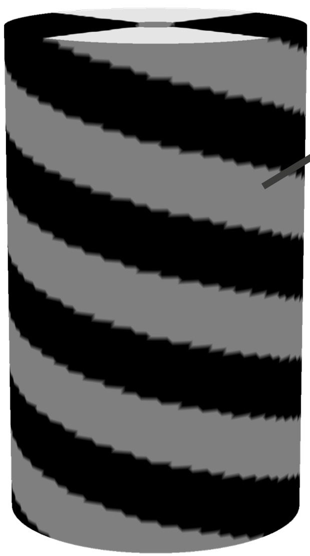


Hydrogen gas corrosion on a metal, cylindrical surface with spiral-shaped channels

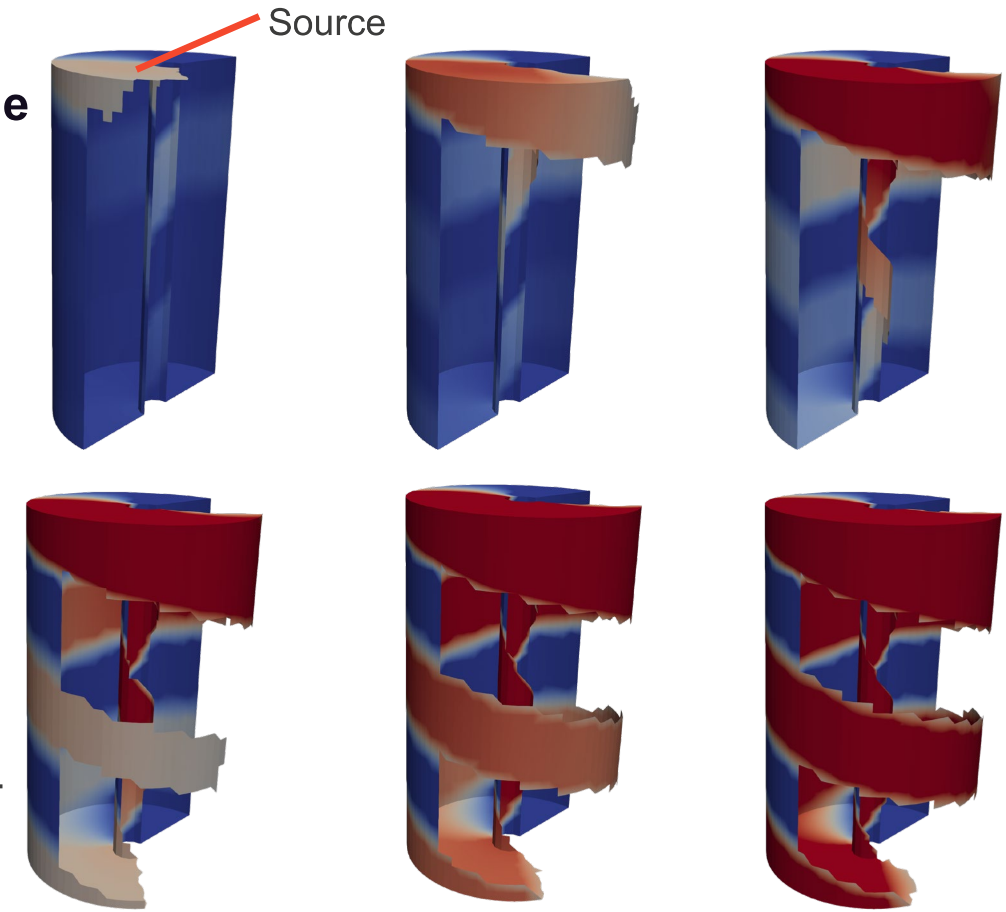
Capabilities: Variable “Skin” Thickness

DRACO can model **surfaces** as meshed 2D manifolds with a, **variable thickness** without the need for a 3D mesh.

Left: A cylindrical surface with spiral-shaped <u>channels</u>.

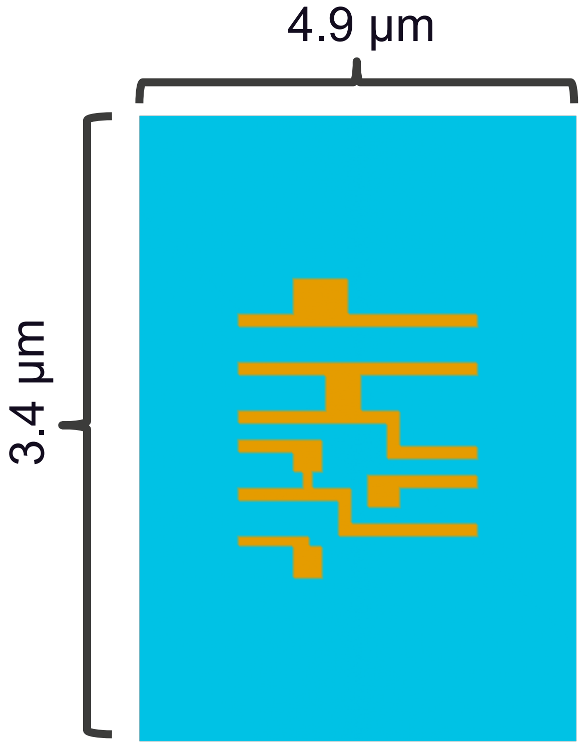


Right: A DRACO simulation of diffusion down one of these channels from a source at the top.



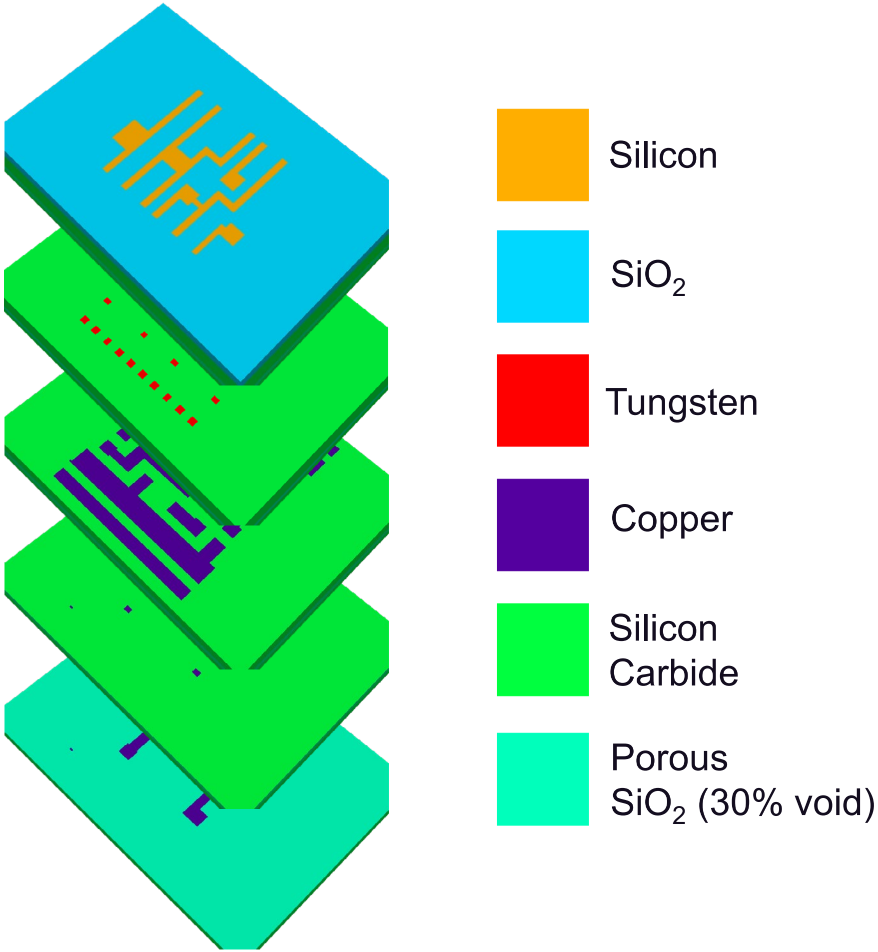
Application: Laser Heating of an Integrated Circuit

Integrated circuit 3D model*:



5 Layers

0.25 µm
Total
Thickness



*IC model data provided by Nina Weisse-Bernstein (ISR-2)
(Laser imaging portion of this work funded by IARPA/RAVEN.)

Application: Laser Heating of an Integrated Circuit

The integrated circuit is illuminated by a 20-100 fs laser pulse. The **excess temperature** *ΔT(z)* deposited by the pulse at a depth *z* is:

Number of Photons in pulse (4×10^6)

Photon Energy (10 keV)

Total Absorption Coefficient

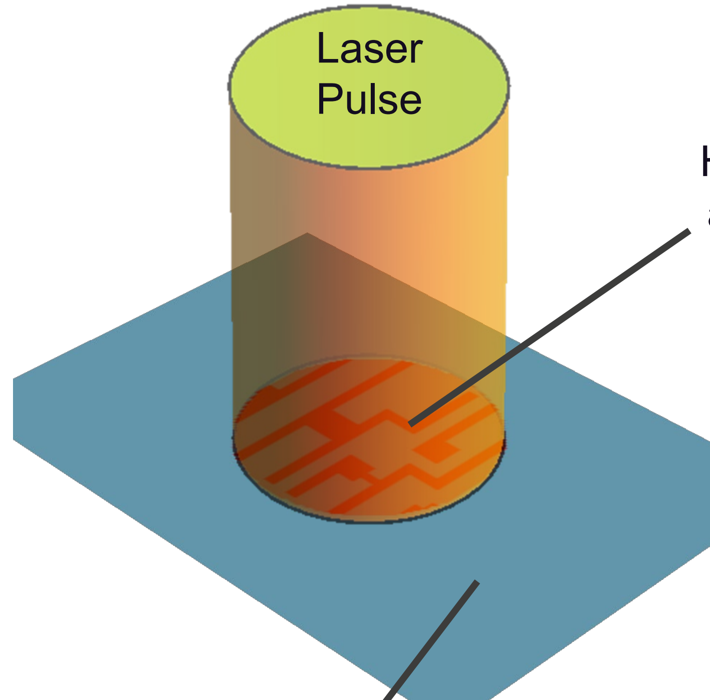
$$\Delta T(z) = \frac{4}{\pi} \frac{N_\gamma E_\gamma}{a^2} \frac{\mu(z)}{C_V(z)} \exp\left(- \underbrace{\int_0^z dz' \, \mu(z')}_{} \right)$$

Spot Diameter (2 μm)

Volumetric Heat Capacity

Line Integral Of Absorption From Sample Surface To Depth *z*

This profile is used as the **initial condition** for a heat diffusion simulation.

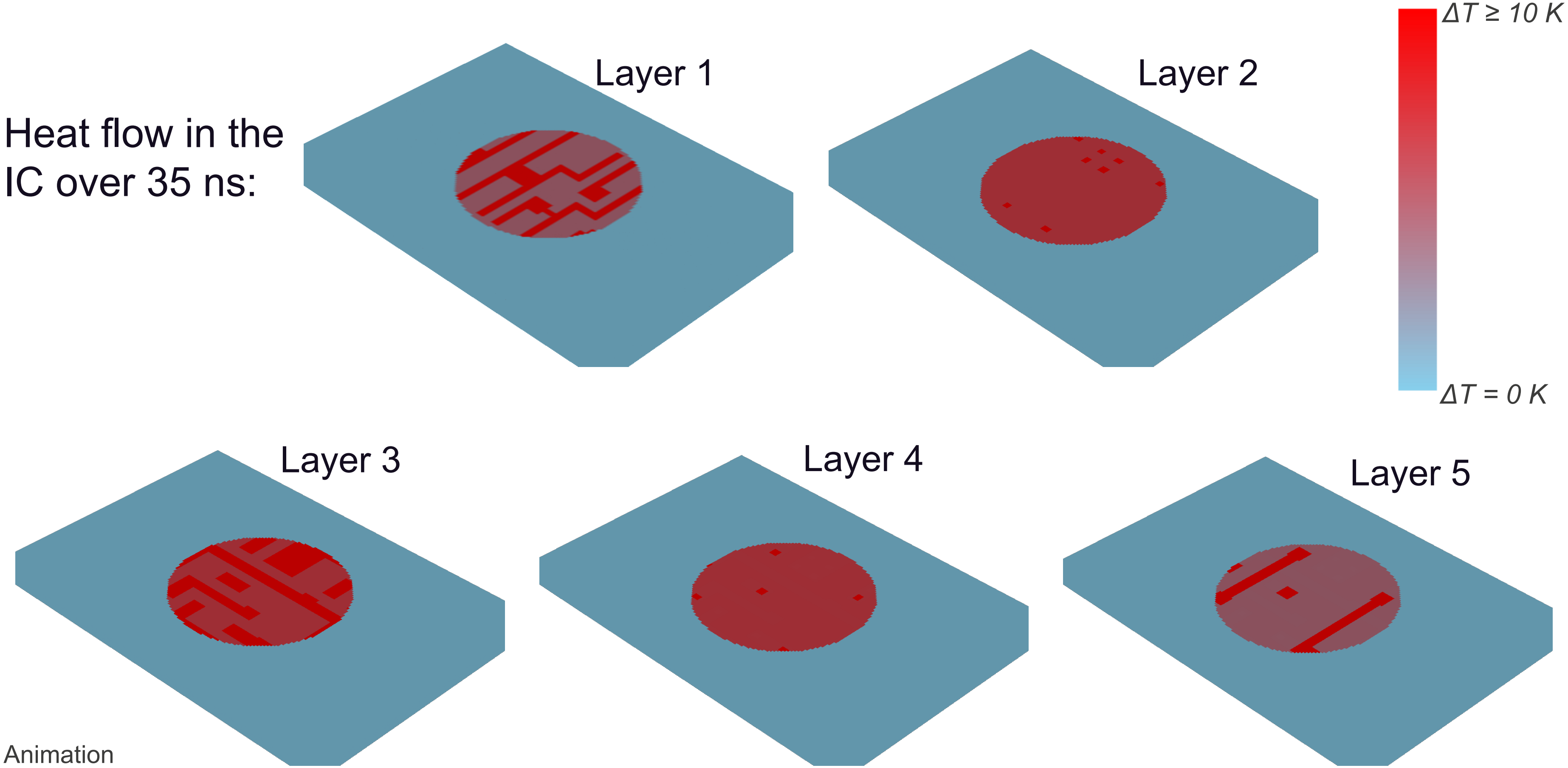


Spot Heating at *t = 0*

Sample remains instantaneously cool around the spot.

Application: Laser Heating of an Integrated Circuit

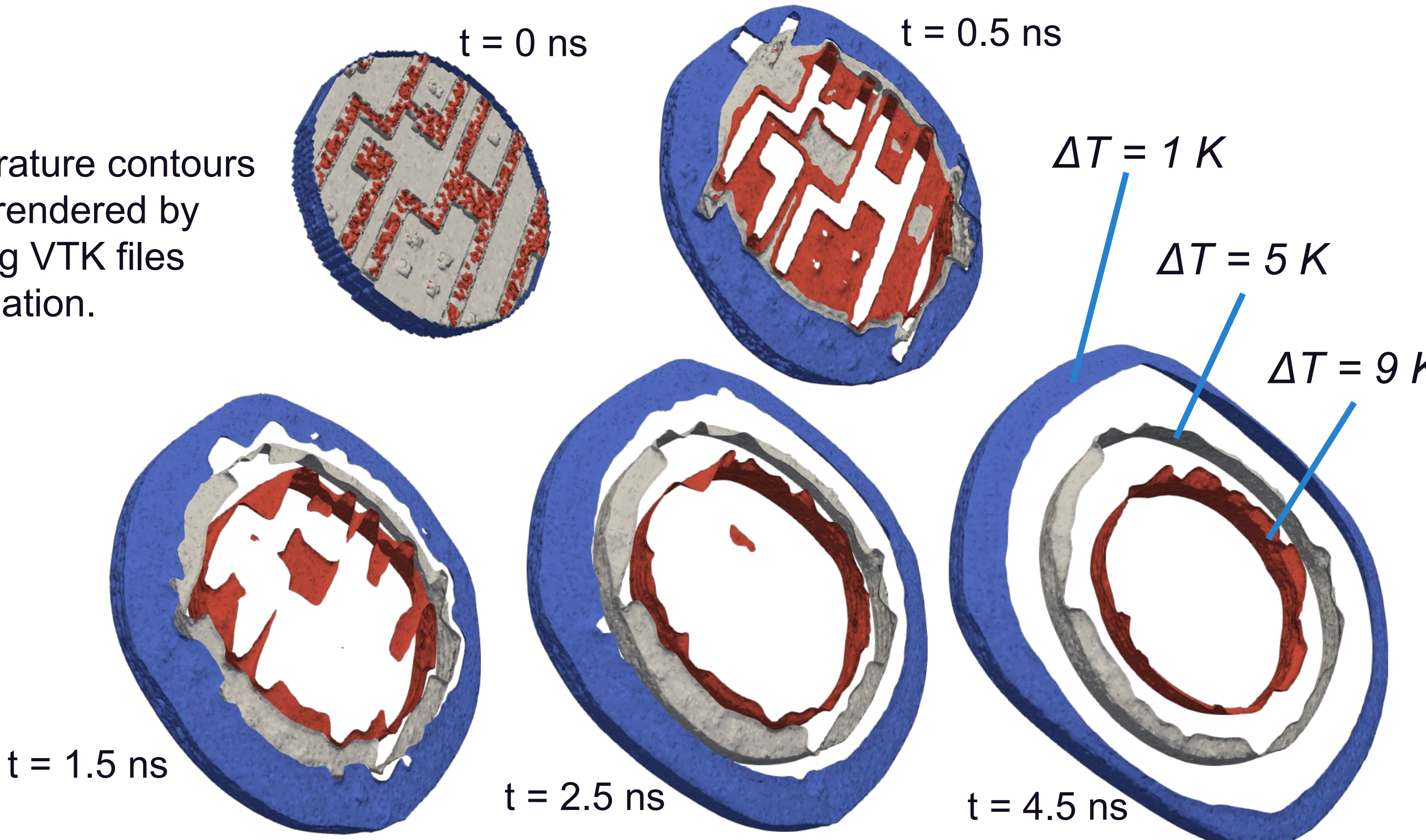
Heat flow in the IC over 35 ns:



Animation

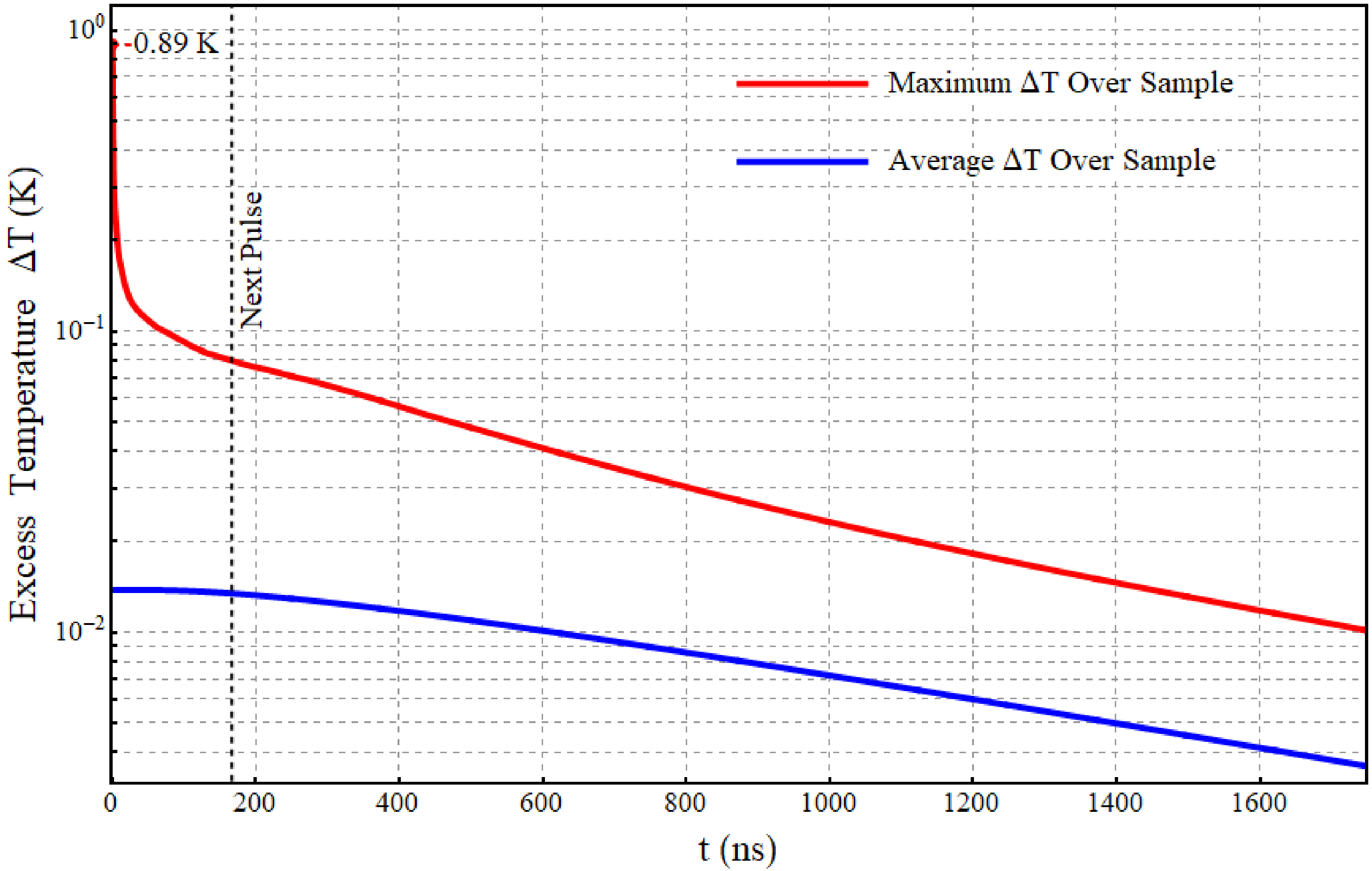
Application: Laser Heating of an Integrated Circuit

Excess temperature contours in the sample rendered by ParaView using VTK files from this simulation.



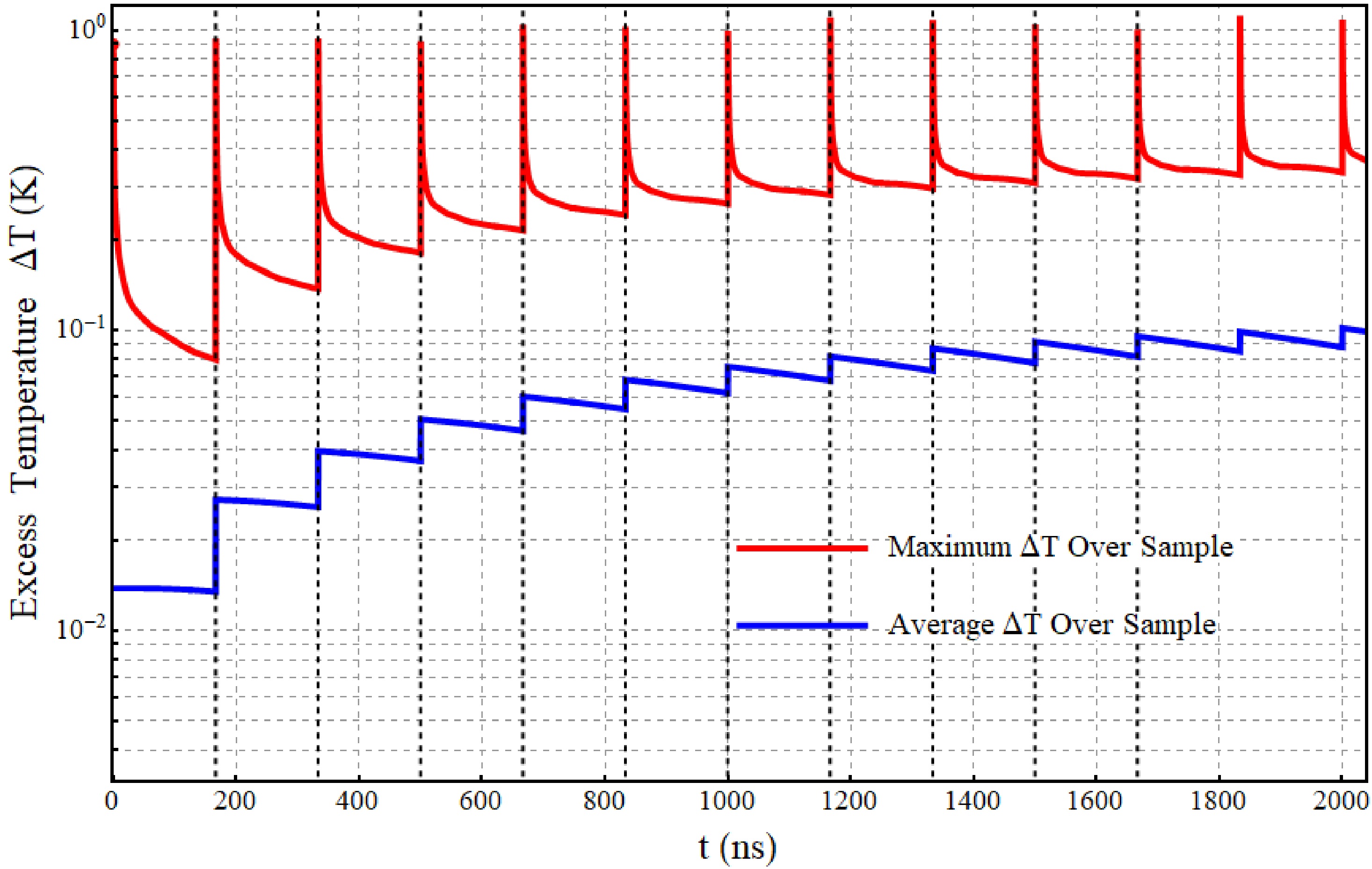
Application: Laser Heating of an Integrated Circuit

Single Laser Pulse



Application: Laser Heating of an Integrated Circuit

Multiple Laser Pulses (Frequency 6 MHz)



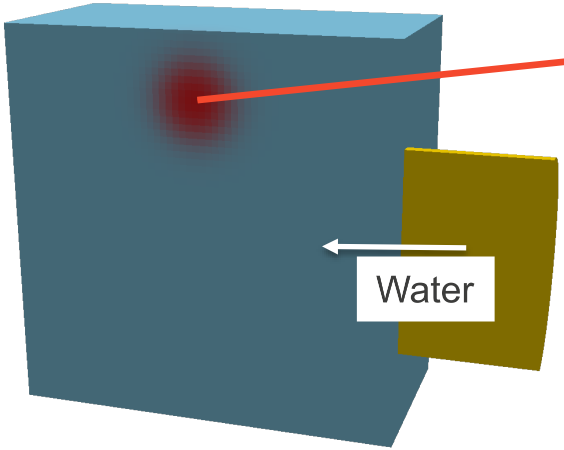
Application : Hydrolysis in HE Binder

The model of Salazar, et. al* says that the polymer HE binder Estane degrades according to the following hydrolysis chemistry:

$$E + A + W \longrightarrow A_2 + L$$
$$A_2 \longrightarrow 2A$$

E = Ester link
A = Acid-ended polymer
W = Water
A_2 = Acid dimer
L = Alcohol-ended polymer

An example DRACO simulation of this chemistry + diffusion consists of a block of HE adjacent to a cylindrical part from which water diffuses into the HE.



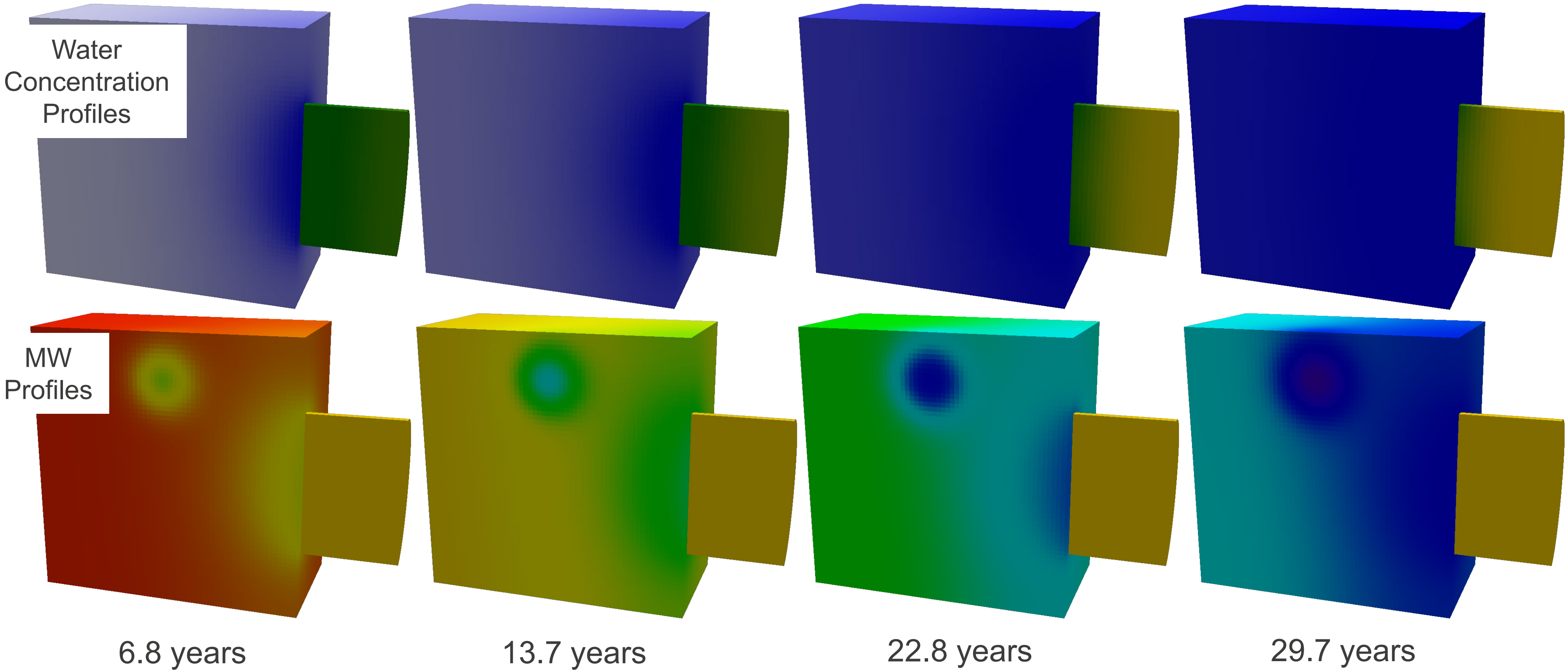
Hot Spot (Increases reaction rate)

Water is generated here and diffuses into the HE

* Salazar, M. & Pack, R. *J. Polym. Sci. B Polym. Phys.*, 40:192-200 (2002)

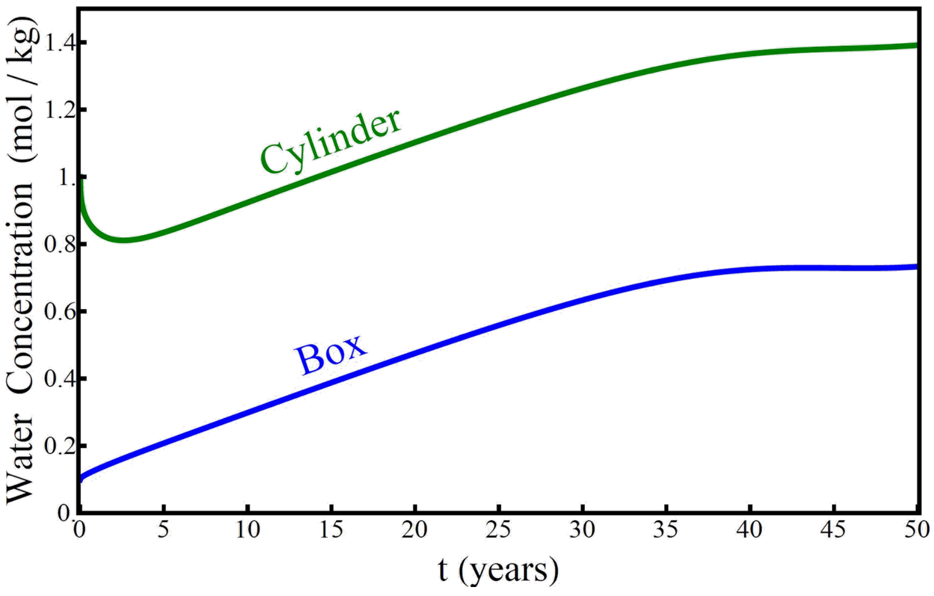
Application : Hydrolysis in HE Binder

Over time, as water diffuses into the HE, this chemistry breaks down the polymer, reducing its mean molecular weight (MW) and reducing mechanical strength.

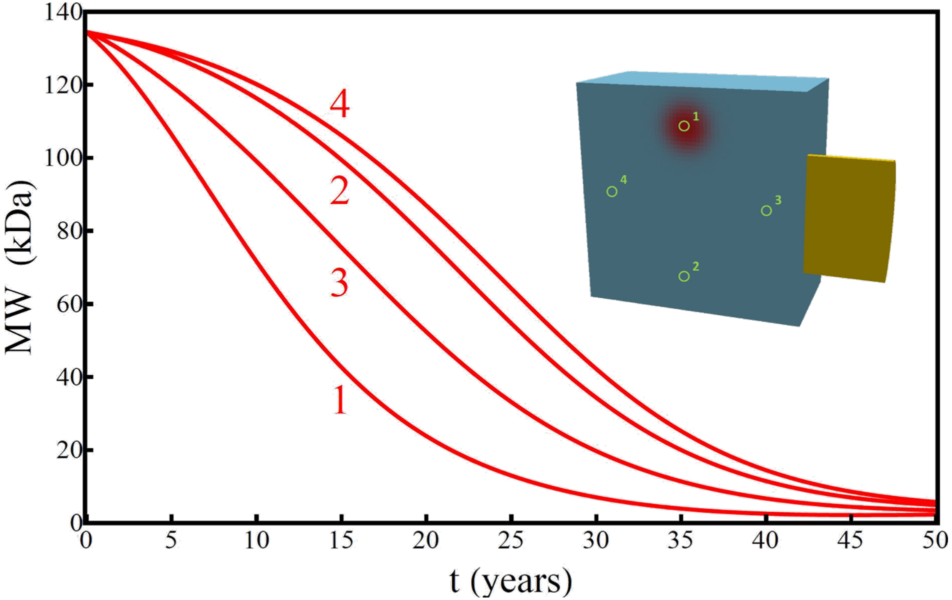


Application : Hydrolysis in HE Binder

Average water concentration in each part over time:



Molecular weight at sampled locations:



Application : Clock Reaction

The **Belousov-Zhabotinsky** (BZ) chemical reaction is an example of a “clock reaction” in which chemical concentrations vary periodically in time. In the presence of diffusion, **diffusion waves** form which propagate across the system.

The simplified “Brusselator model*” of the BZ reaction reduces its complex chemical kinetics to:

$$A \leftrightarrow X$$

$$2X + Y \leftrightarrow 3X$$

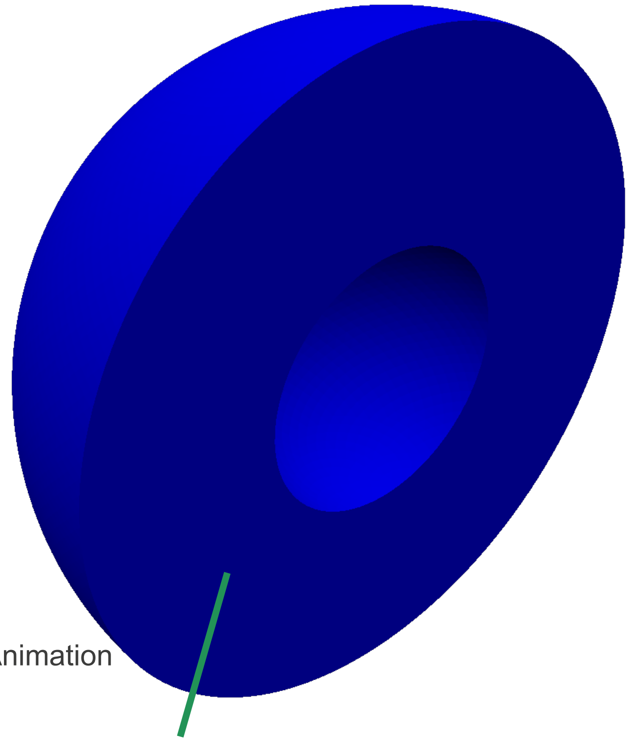
$$B + X \leftrightarrow Y + D$$

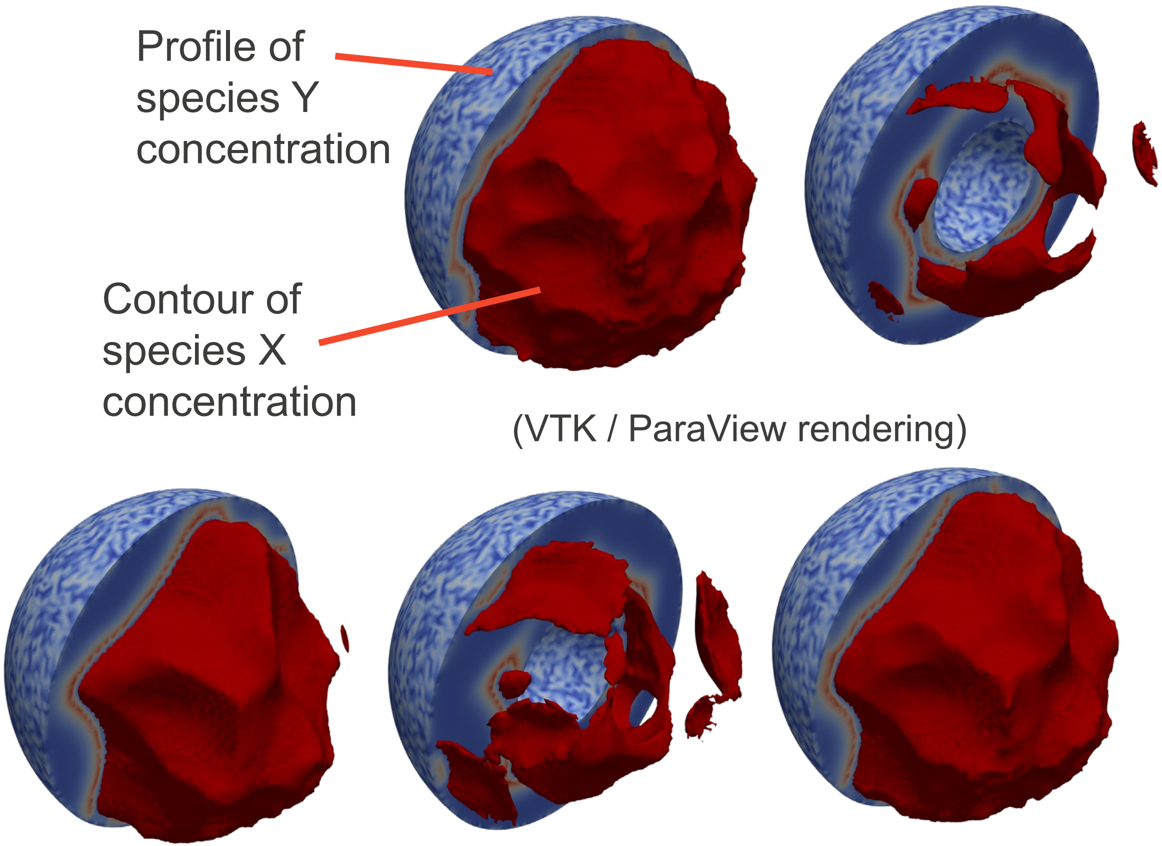
$$X \leftrightarrow E$$

* Prigogine

Application : Clock Reaction

DRACO simulation of the BZ reaction in a spherical shell with diffusion:





(VTK / ParaView rendering)

Summary

- DRACO is a flexible diffusion and chemistry code designed to work on arbitrary, unstructured meshes in 3D.

- It runs in parallel with good scaling.

- DRACO has been designed to be as general as is reasonably possible, and allows the user to specify chemistry, diffusion coefficients, initial conditions, boundary conditions, geometry, mesh resolution, etc.

- DRACO has a number of different types of output, including ray-traced images, VTK 3D rendering files, checkpoints, concentration histories at sampled points or over the entire system, and many others I haven't mentioned.

- I'm always looking for new and interesting areas where DRACO might be applied, as well as for improvements that could be made to DRACO's algorithm and capabilities.